Recovery Models Made SIMPLE

Kalen Delaney
www.SQLServerInternals.com

Kalen Delaney

Background:

- MS in Computer Science from UC Berkeley
- Working exclusively with SQL Server for over **28** years
- Contracted by both Sybase and Microsoft to develop and teach internals courses to Tech Support staff

- Author: *SQL Server Internals: In-memory OLTP* (Red Gate, 2014)
- Primary Author*: SQL Server 2012 Internals* (O'Reilly, 2013)
- Author*: SQL Server Concurrency* (Red Gate, 2011)
- Primary Author*: SQL Server 2008 Internals* (MS Press, 2009)
- Primary Author: *Inside SQL Server 2005* (MS Press, 2007)
- *SQL Server Magazine* columnist and contributing editor
- Editor for Red-Gate: *SQL Server Stairways*

What is SIMPLE Recovery

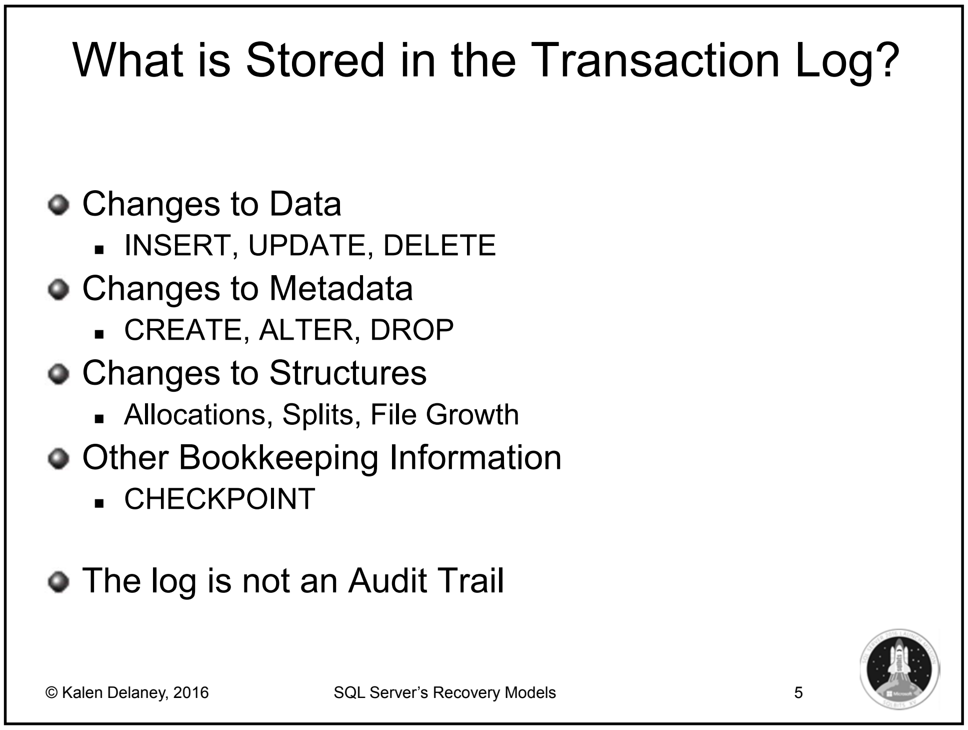
Session Description:

There is a common misconception that SIMPLE Recovery means no logging, and this is a very dangerous myth to propagate. SQL Server does log database changes in SIMPLE Recovery, but some (not all) operations are minimally logged.

In this session, I'll discuss what exactly minimal logging means and what the benefits and dangers of the SIMPLE Recovery Model are.

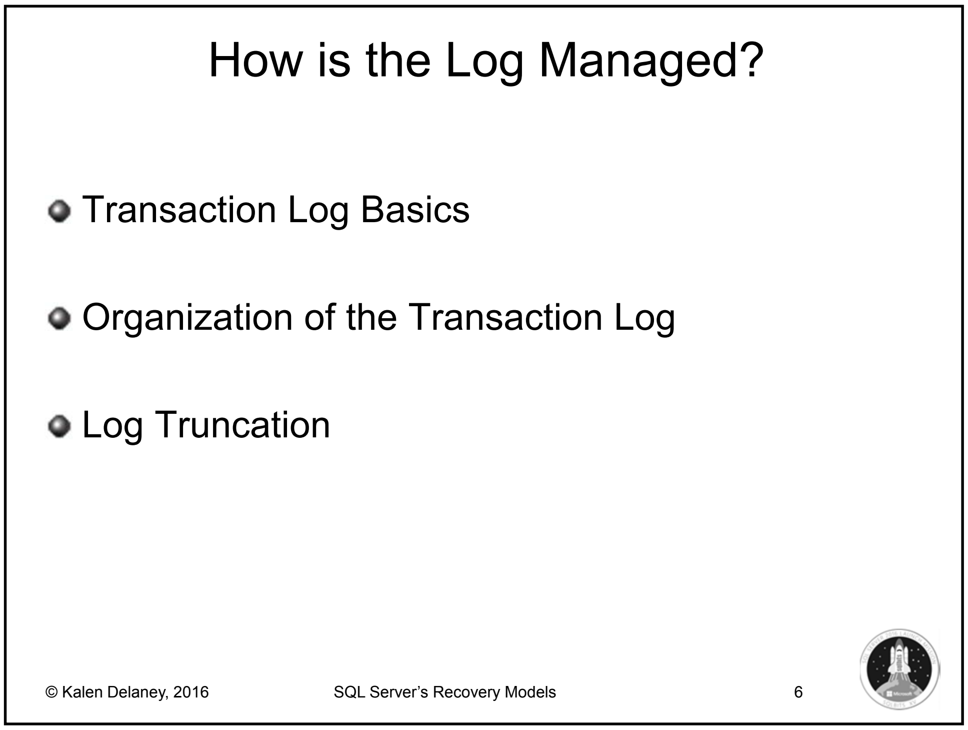
Main Topics

- What is the Transaction Log used for?
- How is the Log Managed?
- What are Recovery Models



What is Stored in the Transaction Log?

- Changes to Data
  - INSERT, UPDATE, DELETE
- Changes to Metadata
  - CREATE, ALTER, DROP
- Changes to Structures
  - Allocations, Splits, File Growth
- Other Bookkeeping Information
  - CHECKPOINT
- The log is not an Audit Trail



How is the Log Managed?

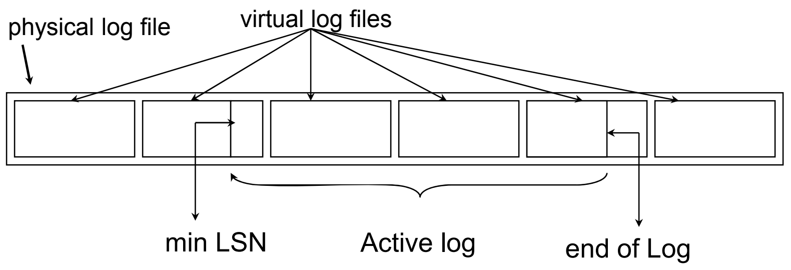
- Transaction Log Basics
- Organization of the Transaction Log
- Log Truncation

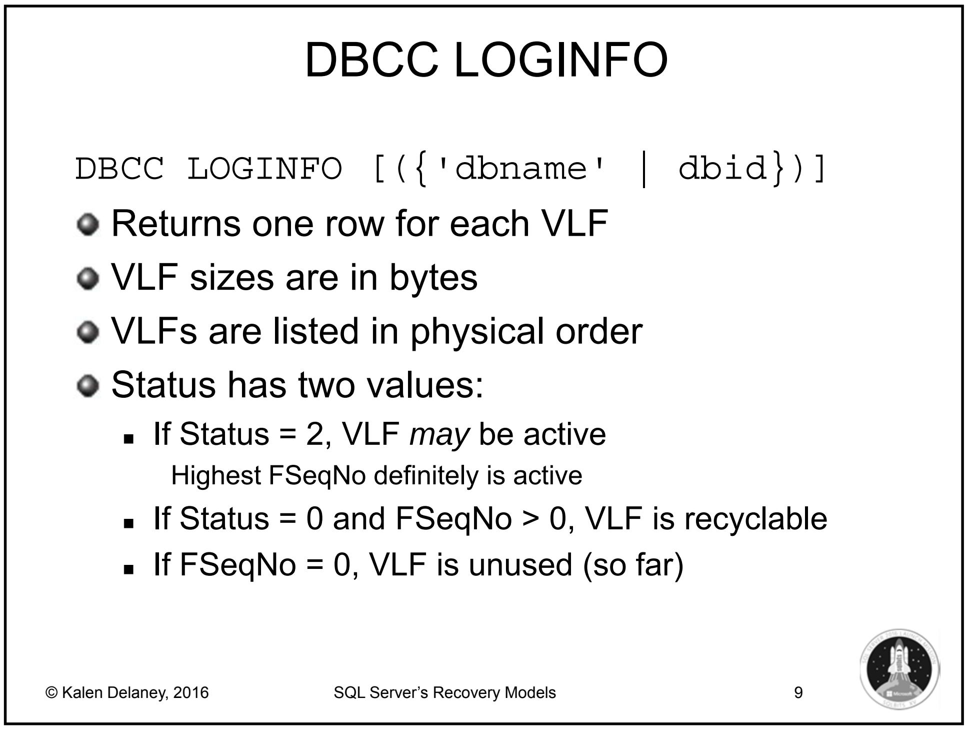
Transaction Log Basics

- Simple Design
  - Log records are not maintained in a system table
  - No Allocation of Pages is Needed
  - Buffer pool not used
  - Log records appended to end of log file
  - Log Records Up to 24K
- Virtual Log Files
  - Used as unit of growth, shrinking and truncation
- LSN - Log Sequence Number
  - Unique identifier for each log record

Organization of Transaction Log

- Log Is Divided Into Virtual Log Files (VLFs)
- VLF Can Be in one of 4 States:
  1. Active
  2. Inactive and not backed up (or not truncated) - recoverable
  3. Inactive and backed up or truncated – recyclable!
  4. Unused
- DBCC LOGINFO Returns One Row Per VLF

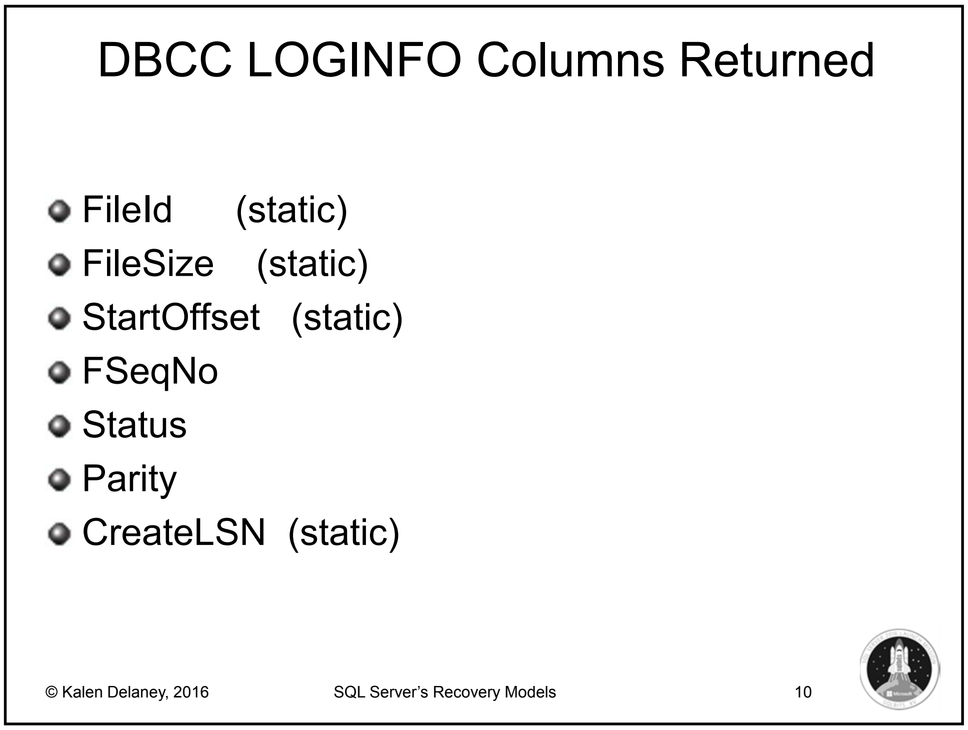




DBCC LOGINFO

```
DBCC LOGINFO [({'dbname' | dbid})]
```

- Returns one row for each VLF
- VLF sizes are in bytes
- VLFs are listed in physical order
- Status has two values:
  - If Status = 2, VLF *may* be active
    - Highest FSeqNo definitely is active
  - If Status = 0 and FSeqNo > 0, VLF is recyclable
  - If FSeqNo = 0, VLF is unused (so far)



DBCC LOGINFO Columns Returned

- FileId (static)
- FileSize (static)
- StartOffset (static)
- FSeqNo
- Status
- Parity
- CreateLSN (static)

DEMO #1

DBCC LOGINFO

Log Truncation

- Truncation is a **Logical** Operation
- Non-active VLF(s) marked as Reusable
- Occurs:
  - When log is backed up
    - Prior to SQL Server 2008:
      BACKUP LOG … WITH TRUNCATE_ONLY
  - When recovery model is changed to SIMPLE
  - At Checkpoint, in auto-truncate mode
    - When DB is in SIMPLE recovery model
    - No Full DB Backup has been made
      - since last time DB was in SIMPLE recovery
      - or ever!

DEMO #2

Log Truncation

Recovery Models

- Recovery Models Overview
- Minimally Logged Operations
- FULL Recovery Model
- BULK_LOGGED Recovery Model
- SIMPLE Recovery Model
- Choosing a Recovery Model

Recovery Models Overview

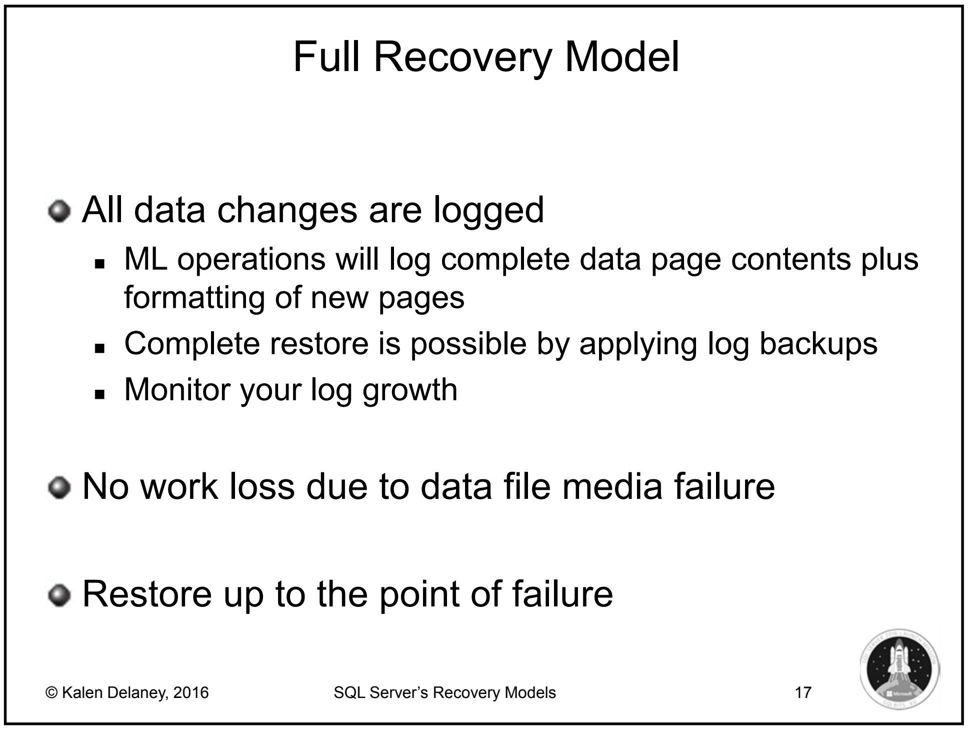
- Three Types
  - Full
  - Bulk_Logged
  - Simple
- Set With ALTER DATABASE
- Tradeoffs:
  - Speed of certain operations
  - Size of log
  - Size of log backup
  - Recoverability

Minimally Logged Operations (ML)

Certain Operations do not log every individual row

Minimum requirement is enough information needed to rollback

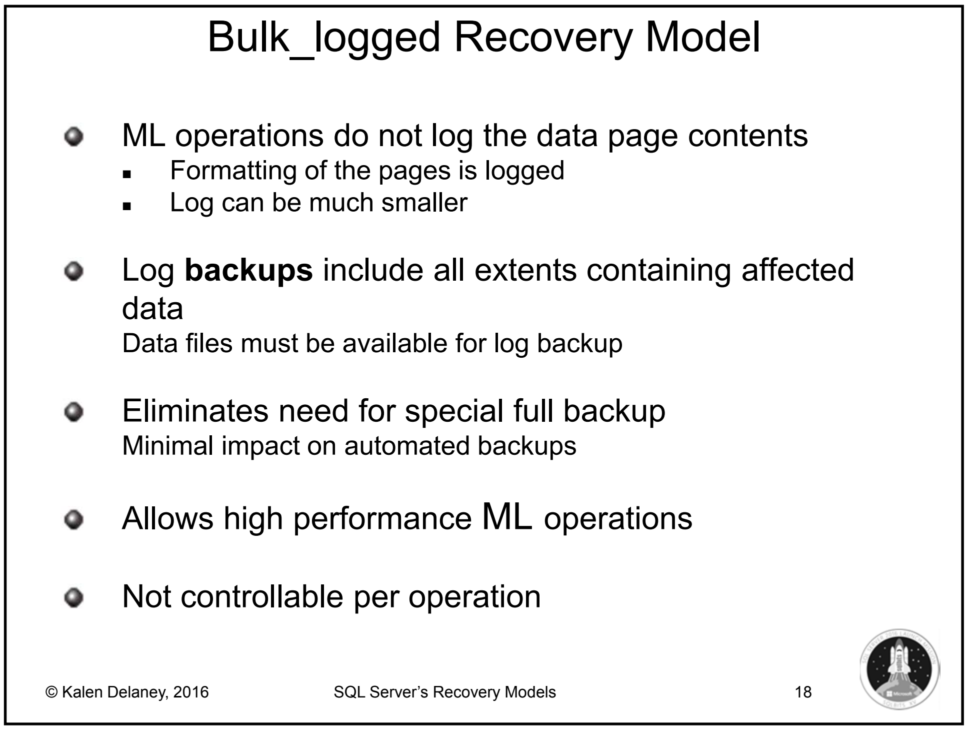
- SELECT INTO
- Bulk Import: BULK INSERT, bcp
- CREATE/ALTER/DROP INDEX
- INSERT INTO … SELECT (2008 and later)
- Tables that are replicated or enabled for CDC will have every row logged



Full Recovery Model

- All data changes are logged
  - ML operations will log complete data page contents plus formatting of new pages
  - Complete restore is possible by applying log backups
  - Monitor your log growth
- No work loss due to data file media failure
- Restore up to the point of failure

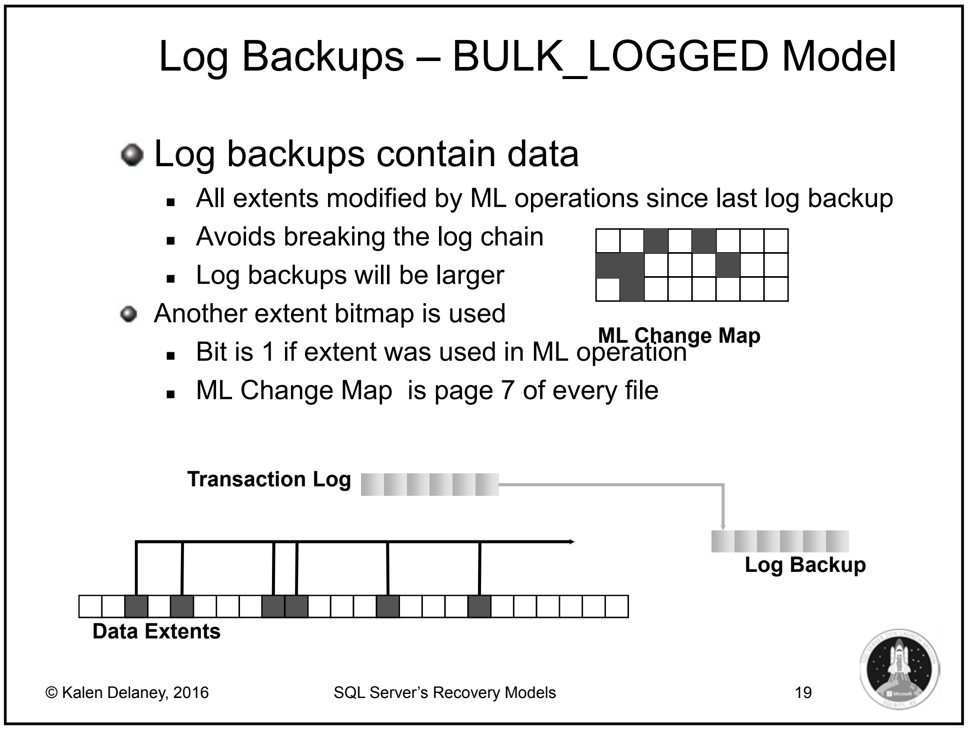
© Kalen Delaney, 2016 SQL Server's Recovery Models 17



Bulk_logged Recovery Model

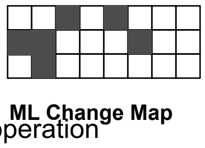
- ML operations do not log the data page contents
  - Formatting of the pages is logged
  - Log can be much smaller
- Log **backups** include all extents containing affected data
  Data files must be available for log backup
- Eliminates need for special full backup
  Minimal impact on automated backups
- Allows high performance ML operations
- Not controllable per operation

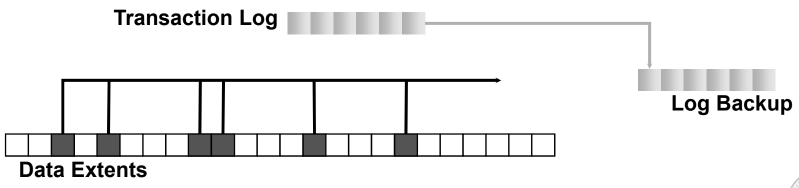
© Kalen Delaney, 2016 SQL Server's Recovery Models 18

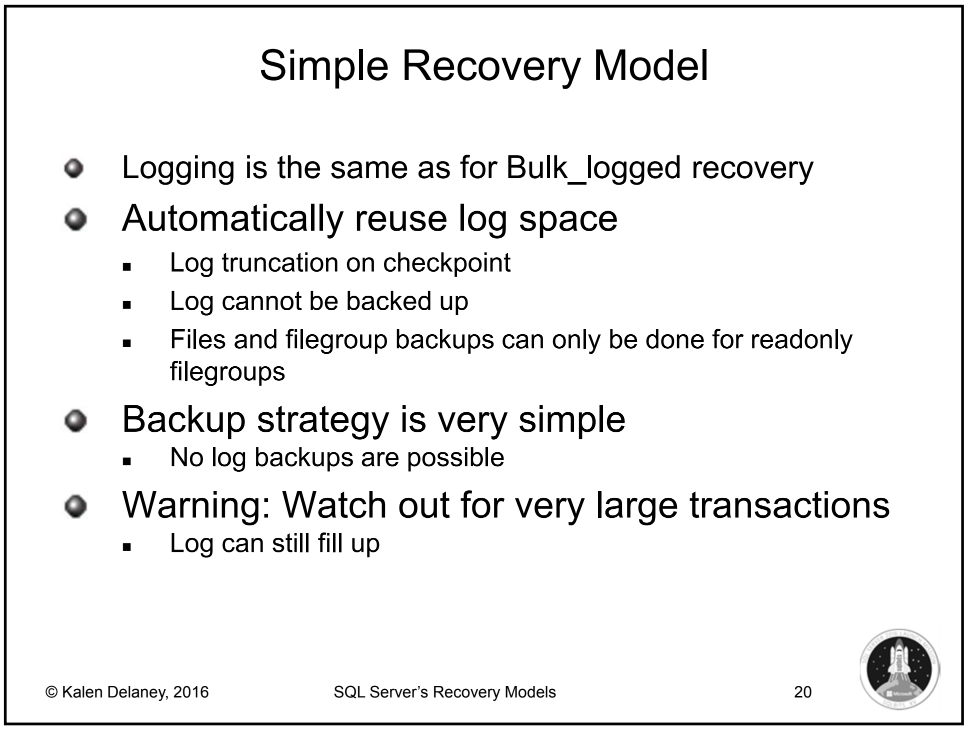


Log Backups – BULK_LOGGED Model

- Log backups contain data
  - All extents modified by ML operations since last log backup
  - Avoids breaking the log chain
  - Log backups will be larger
- Another extent bitmap is used
  - Bit is 1 if extent was used in ML operation
  - ML Change Map is page 7 of every file







Simple Recovery Model

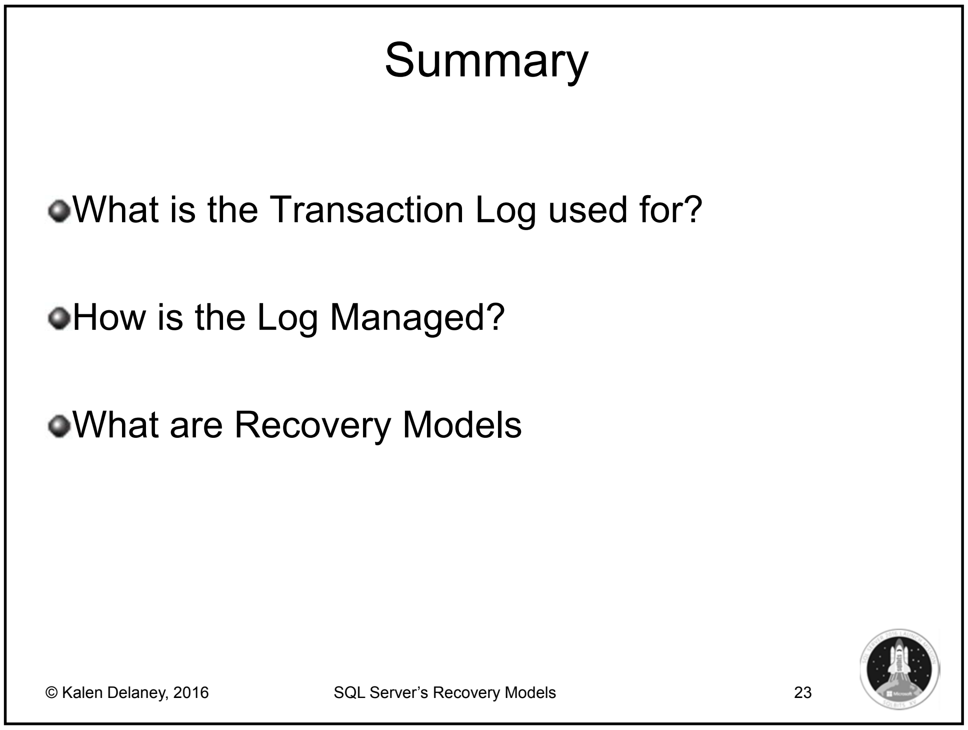
- Logging is the same as for Bulk_logged recovery
- Automatically reuse log space
  - Log truncation on checkpoint
  - Log cannot be backed up
  - Files and filegroup backups can only be done for readonly filegroups
- Backup strategy is very simple
  - No log backups are possible
- Warning: Watch out for very large transactions
  - Log can still fill up

DEMO #3

Log space used in each recovery model

Choosing a Recovery Model

- First, determine your backup strategy
  - If you'll be making regular log backups, then FULL
  - If there will be no log backups, then SIMPLE
- Switch to BULK_LOGGED when necessary
  - ML operations are faster
  - Log backup made after ML operation cannot be restored to a point in time
  - Tail of log cannot be backed up if ML data is on damaged file



Summary

- What is the Transaction Log used for?
- How is the Log Managed?
- What are Recovery Models

© Kalen Delaney, 2016 SQL Server's Recovery Models 23

Thank You!

© Kalen Delaney, 2016 SQL Server's Recovery Models 24